1B

УДК 004.8

О.О. Варламов
Московский автомобильно-дорожный государственный
технический университет (МАДИ),
Московский физико-технический институт (государственный университет),
НИИ «МИВАР», г. Москва
Россия, 127572, г. Москва, Алтуфьевское ш., 95«Б», оф. 41
ovar@narod.ru; OVarlamov@gmail.com

Миварные технологии: переход от продукций к двудольным миварным сетям и реализация автоматического конструктора алгоритмов, управляемого потоком входных данных и обрабатывающего более трех миллионов правил

O.O. Varlamov
Moscow Automobile and Road State Technical University (MADI),
Moscow Institute of Physics and Technologies (university) (MIPT),
Research Institute «MIVAR», Moscow
Russia, 127572, Moscow, Altufievskoe highway, 95«B», of. 41
ovar@narod.ru; OVarlamov@gmail.com

MIVAR Technologies: Transition from Productions to Bipartite Graphs MIVAR Nets and Realization of Automated Constructor of Algorithms Handling More than Three Million Production Rules

О.О. Варламов
Московський автомобільно-дорожній державний технічний університет (МАДІ),
Московський фізико-технічний інститут (державний університет),
НДІ «МІВАР», м. Москва
Росія, 127572, м. Москва, Алтуф’євське ш., 95 «Б», оф. 41
ovar@narod.ru; OVarlamov@gmail.com

Міварні технології і перехід від продукції до дводольних міварних мереж та реалізація автоматичного конструктора алгоритмів, яким керує потік вхідних даних і обробляє більше трьох мільйонів правил

Теоретически обоснована линейная вычислительная сложность автоматического конструирования алгоритмов (логического вывода) из переменных-объектов и правил-процедур миварных сетей. В качестве миварных правил могут быть использованы сервисы, модули и вычислительные процедуры. На основе миварных сетей создан программный комплекс УДАВ, который обрабатывает более 1,17 млн переменных и более 3,5 млн правил на обычных компьютерах и ноутбуках. Приведены результаты экспериментов, которые подтверждают линейную сложность логического вывода в формализме миварных сетей.
Ключевые слова: мивар, миварные сети, логический вывод, вычислительная сложность, искусственный интеллект, интеллектуальные системы, экспертные системы.

The theoretical transition from the graphs of production systems to the bipartite graphs of the MIVAR nets is shown. Examples of the implementation of the MIVAR nets in the formalisms of matrixes and graphs are given. The linear computational complexity of algorithms for automated building of objects and rules of the MIVAR nets is theoretically proved. On the basis of the MIVAR nets the UDAV software complex is developed, handling more than 1.17 million objects and more than 3.5 million rules on ordinary computers. The results of experiments that confirm a linear computational complexity of the MIVAR method of information processing are given.
Key words: MIVAR, MIVAR net, logical inference, computational complexity, artificial intelligence, intelligent systems, expert systems.

Теоретично обґрунтована лінійна обчислювальна складність автоматичного конструювання алгоритмів (логічного висновку) зі змінних-об'єктів та правил-процедур міварних мереж. Як міварні правила можуть бути використані сервіси, модулі та обчислювальні процедури. На основі міварних мереж створений програмний комплекс «УДАВ», що оброблює більше 1,17 млн змінних та більш ніж 3,5 млн правил на звичайних комп'ютерах та ноутбуках. Наведені результати експериментів, які підтверджують лінійну складність логічного висновку у формалізмі міварних мереж.
Ключові слова: мівар, Міварні мережі, логічний висновок, обчислювальна складність, штучний інтелект, інтелектуальні системи, експертні системи.

Введение

Проблема создания интеллектуальных систем остается актуальной и практически значимой. Создание экспертных систем нового поколения позволит автоматизировать решение различных сложных интеллектуальных задач и повысит конкурентоспособность своих пользователей. Термин МИВАР – это сокращение от названия «Многомерная информационная варьирующаяся адаптивная реальность». Миварный подход позволил предложить новые модели и методы обработки информации и управления [1-22]. Миварные технологии накопления и обработки информации разрабатываются в России достаточно давно. Первые статьи были посвящены исследованию некоторых задач теории графов и разработке линейного матричного метода определения маршрута логического вывода на адаптивной сети правил [1-3]. Затем были работы по созданию миварного информационного пространства и эволюционных баз данных и правил [4], [5]. Наиболее строгое формализованное и теоретическое оформление мивары получили в работах [6], [7]. Затем были рассмотрены вопросы развития миваров [8-10] и их применения для создания различных тренажеров и обучающих систем [11-23].

Будем понимать под системами искусственного интеллекта активные самообучающиеся логически рассуждающие системы. В прошлом веке были разработаны технологии создания экспертных систем по отдельным узконаправленным предметным областям. Это было обусловлено сложностями формализованного описания требуемых предметных областей и тем, что системы логического вывода не могли обрабатывать на практике более 20 объектов/правил. В то же время получили развитие «интеллектуальные пакеты прикладных программ» (ИППП), которые позволяли решать в автоматизированном режиме задачи в разных областях, где требовались вычисления и конструирование алгоритмов решения задач. Технологии ИППП развиваются в миварах.

Миварный подход объединяет и развивает достижения в: базах данных, решении вычислительных задач, логической обработке. Он включает 2 основные технологии.

1. *Миварная технология накопления информации* – это способ создания глобальных эволюционных баз данных и правил (знаний) с изменяемой структурой на основе многомерного адаптивного дискретного миварного информационного пространства унифицированного представления данных и правил, базирующегося на трех основных понятиях «вещь, свойство, отношение».

2. *Миварная технология обработки информации* – это способ создания системы логического вывода или «автоматического конструирования алгоритмов из модулей, сервисов или процедур» на основе активной обучаемой миварной сети правил с линейной вычислительной сложностью.

Миварная технология накопления информации предназначена для хранения любой информации с возможным эволюционным изменением структуры и без ограничений по объему и формам представления. Миварная технология обработки информации предназначена для обработки информации, включая логический вывод, вычислительные процедуры и «сервисы». Фактически, миварные сети позволяют развить продукционный подход и создать автоматическую обучаемую логически рассуждающую систему. Миварный подход развивает продукционные системы, онтологии, семантические сети, сервисно-ориентированные архитектуры, многоагентные системы и другие современные информационные технологии в целях создания интеллектуальных систем и систем ИИ.

Анализ парадигм и моделей обработки данных

Традиционно выделяют следующие парадигмы и модели обработки данных: исчисление высказываний, исчисление предикатов, продукции, семантические сети, онтологии и др. У продукционного подхода есть важные преимущества. Д.А. Поспелов писал, что знания о внешнем мире могут иметь двоякую природу:

– могут содержать декларативное описание фактов и явлений внешнего мира, фиксирующее их наличие или отсутствие, а также основные связи и закономерности, в которые эти факты и явления входят;

– могут содержать и процедурные описания того, как надо манипулировать с этими фактами и достигать целей, интересных для системы [24].

Продукции в общем виде записывают в форме «если... то...». Часть специалистов по интеллектуальным системам считает, что запись знаний в виде систем продукций носит универсальный характер – любые знания можно записать в такой форме. В системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. К ним, по сути, сводятся все каузальные, т.е. причинно-следственные утверждения. Для решения многих практических задач применения информационных систем и процессов требуется проводить как логическую, так и вычислительную обработку данных. Исторически так сложилось, что области логического вывода и вычислительной обработки развивались самостоятельно и решали различные классы задач. В некотором смысле, существовало противоречие между этими подходами [2-4], [10-17], [24-33]. Кроме того, разделяли проблемы обработки и хранения различных данных. Базы данных преимущественно использовались только для хранения и поиска требуемых данных, а системы логического вывода и вычислений применялись для обработки информации, поиска решений и т.п. Получалось, что эти области относительно слабо пересекались, хотя, в плане перспектив развития в каждой из них регулярно провозглашались цели объединения всех функций по накоплению и обработке информации в одной системе [4], [10], [24-33].

Если проводить аналогию с человеком, то наш разум одновременно накапливает и хранит данные, комплексно решает и логические, и вычислительные, и логико-вычислительные задачи. Миварный подход позволяет в едином формализме проводить и эволюционное накопление данных в миварном информационном пространстве, и выполнять совмещенную логико-вычислительную обработку в миварных сетях.

Возможности и ограничения продукционного подхода

В настоящее время продолжается дискуссия о роли и возможных применениях различных логических механизмов и, прежде всего, исчисления высказываний и исчисления предикатов и продукций. У продукционного подхода есть важные преимущества. Приведем мнение Д.А. Поспелова: «Для описания знаний в интеллектуальных системах используются специальные языки описания знаний (ЯОЗ)... Простейшими видами таких ЯОЗ являются языки исчисления высказываний или исчисления предикатов вместе с теми процедурами вывода, которые для них известны. Однако в современных интеллектуальных системах такие языки используются довольно редко. Куда более распространены в них языки, основанные на продукциях.... Продукции в общем виде можно записать в форме «если..., то...», но к продукциям относятся не только выражения, имеющие эту форму, но и многие другие» [24, с. 129]. У Д.А. Поспелова приведены 9 типов продукций и специально подчеркнуто, что возможны продукции и других типов [24, с. 131-134]. Далее делается вывод: «...продукции могут иметь весьма различное значение. В качестве их левых и правых частей могут выступать и некоторые утверждения, и действия» [24, с. 134]. Отметим, что О.П. Кузнецов в [25, с. 282-283] под продукциями понимает множество правил вывода в канонических системах (системах продукций Поста), в которых есть посылки и следствия. С точки зрения анализа миварного метода обработки данных принципиально важным является то, что в системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. Следовательно, применение продукционного подхода для логико-вычислительной обработки разнообразных данных является целесообразным.

Д.А. Поспелов дал следующее определение: «Продукционной системой будем называть любую совокупность продукций, в которую могут входить продукции любого из перечисленных выше типов» [24, с. 134]. Существуют различные конструкции продукций. В наиболее общем виде «вместо продукций типа $a => b$ рассматривают более сложные конструкции. В общей форме продукции имеют вид: $i, П, P, A => B, Q$.

Здесь $A => B$ – это обычная продукция «если... то...», которая носит название ядра продукции. Элемент P характеризует внешние условия или условия применимости продукции, определяемые факторами, не входящими непосредственно в A, например, целями, которые стоят перед рассуждающей системой. Условия P позволяют из всех продукций, у которых в левой части ядра стоит A, отбирать нужную часть продукций. Элемент $П$ характеризует сферу проблемной области базы знаний, или предусловия применимости продукции. Эти предусловия ничем не отличаются от P, но выделяют подсистемы продукций на ранг выше тех, которые выделяют условия. Предусловия задают формальную систему, в рамках которой будут проводиться логические рассуждения... Наконец, Q характеризует постусловия продукции, указывающие на те изменения, которые необходимо внести в базу знаний и в систему продукций после реализации данной продукции» [24, с. 134-135].

Под логической обработкой принято понимать некий вывод, лежащий в основе человеческих рассуждений. Для проведения анализа используем описания и исходную информацию из широко известных источников. В работе Д.А. Поспелова выдвинуто следующее основополагающее положение: «Всякий вывод, как бы он не был организован, носит переборный характер... Программа вынуждена перебирать варианты, заходить в тупики, проходить циклы прежде, чем она сможет найти правильный путь решения. Повышение эффективности процесса вывода – центральная проблема всех автоматизированных систем дедуктивного вывода» [24, с. 79].

Особый интерес представляет описание общей схемы выводов, лежащей «в основе большого количества моделей человеческих достоверных рассуждений» [24, с. 83]. Д.А. Поспелов пишет: «Схема вывода не обязательно описывается в виде дерева. Она может иметь вид произвольной сети, ориентированной, неориентированной или частично ориентированной» [24, с. 84]. На рис. 1 показан пример неориентированной сети, аналогичный рисунку в [24, с. 83].

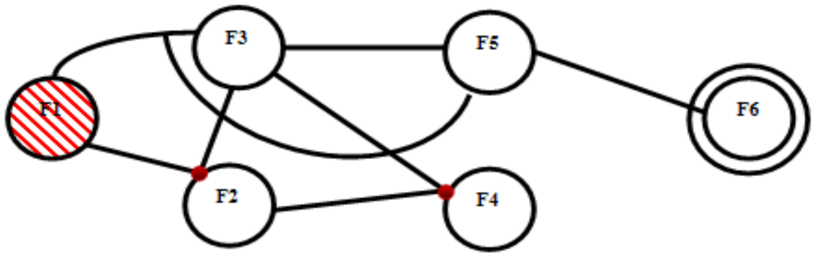


Рисунок 1 – Пример однодольного графа неориентированной «И-ИЛИ сети»

«Такая сеть (наличие или отсутствие ориентации не играет здесь роли) называется И-ИЛИ сетью. Процесс вывода на И-ИЛИ сети протекает следующим образом. Пусть мы хотим доказать утверждение F6. В качестве априорно доказанного задано утверждение F1. Как из F1 можно получить F6? Если считать, что все связи допускают ориентацию в нужную сторону, то из F1 можно получить F3, затем F5 и, наконец, F6. Но этот путь нам удалось отыскать потому, что сеть, показанную на рис. 1, мы видим “с высоты птичьего полета”. Лабиринт поиска лежит в виде чертежа перед нами. Именно это позволяет нам не делать лишних попыток, не двигаться в ненужную сторону, а идти кратчайшим путем к цели» [24, с. 84]. Это выражало общее мнение ученых. Отметим, что на рис. 1 фактически изображен однодольный граф, т.е. все вершины графа принадлежат одному классу [25, с. 125]. Этот факт будет очень важен для дальнейшего анализа.

Д.А. Поспелов описал два метода логического вывода. «Эта процедура известна среди специалистов под названием метода прямой волны. Волна поиска путей к целевой площадке распространяется от всех площадок, играющих роль начальных... Возможен и другой способ поиска доказательства. Он носит название метода обратной волны… Часто используется смешанный метод вывода, при котором одновременно движутся прямая и обратная волны. При встрече этих волн формируется путь вывода от начальных аксиом к целевым выражениям» [25, с. 85].

Отметим, что эти работы не нашли широкого практического применения в прошлом веке. Это обусловлено тем, что не было предложено метода быстрой логической обработки данных. Как было показано выше, ученые исходили из предпосылки, что любой вывод носит переборный характер, что означает факториальный рост вычислительной сложности при увеличении количества правил. Даже современные суперкомпьютеры при таком подходе позволяют обрабатывать в реальном масштабе времени лишь 20 правил. Фактически, это позволяет решать только «игрушечные» модельные задачи, так как современные требования начинаются с сотен и достигают десятков тысяч переменных и правил даже для относительно простых предметных областей. Миварный подход позволил эффективно реализовать предложенные для продукций Д.А. Поспеловым методы прямой и обратной волны [24, с. 85].

Системы искусственного интеллекта и мивары

Рассмотрим основные подходы к созданию искусственного интеллекта (ИИ) и экспертных систем. Приведем мнение Дж. Люгера, который считал «…объединение разрозненных областей искусственного интеллекта с помощью детального описания

его теоретических основ… Интеллект – это сложная область знаний, которую невозможно описать с помощью какой-то одной теории. Ученые строят целую иерархию теорий, характеризующих его на разных уровнях абстракции» [33, с. 20]. Формальный подход Рассела и Уайтхеда к математическим умозаключениям сделал возможной его автоматизацию в реальных вычислительных машинах. Логический синтаксис и формальные правила вывода, разработанные Расселом и Уайтхедом, лежат в основе систем автоматического доказательства теорем и составляют теоретические основы искусственного интеллекта [33, с. 34-35]. Однако, с другой стороны, это наложило и важные ограничения на возможности описания реальных предметных областей: формализм, как выясняется, тоже имеет свои недостатки вместе с преимуществами. Люгер справедливо отмечает, что ИИ не стал жизнеспособной научной дисциплиной до появления цифровых вычислительных машин.

Однако архитектура цифровых компьютеров наталкивает на специфичное представление теории ИИ. Получается, что интеллект – это способ обработки информации. Далее Люгер формулирует замечательную мысль, к которой мы с удовольствием присоединяемся: «Мы часто забываем, что инструменты, которые мы создаем для своих целей, влияют своим устройством и ограничениями на формирование наших представлений о мире». Такое, казалось бы, стесняющее наш кругозор взаимодействие является важным аспектом развития человеческого знания: инструмент (а научные теории, в конечном счете, тоже инструменты) создается для решения конкретной проблемы. По мере применения и совершенствования инструмент подсказывает другие способы его использования, которые приводят к новым вопросам и, в конце концов – разработке новых инструментов [33, с. 35]. Если говорить о миварах и формализмах Фреге, Рассела и Уайтхеда, Тарского и многих других, то надо помнить, что наука развивается по спирали и старые формализмы требуют своего развития на новом витке. Упомянутые выше исчисление предикатов и другие формализмы при необходимости могут быть реализованы и в правилах (отношениях) миварных сетей. Как видно из [24-33], в современной теории ИИ кроме предикатного подхода уже разработано большое количество других подходов, кардинально отличающихся от исчисления предикатов. Подчеркнем, что продукционный подход и его развитие в миварных сетях являются еще одной альтернативой исчислению предикатов при создании ИИ.

В области ИИ существует две фундаментальные проблемы – это представление знаний и поиск [33, с. 42]. Первая проблема относится к получению новых знаний с помощью формального языка. Поиск – это метод решения проблемы, в котором систематически просматривается пространство состояний задачи, т.е. альтернативных стадий ее решения. Дж. Люгер подтверждает наш тезис о том, что в прошлом веке даже в самых передовых областях ИИ ученые не смогли решить сложные задачи [33, с. 44], а значит, нужны принципиально новые подходы и исследования, к числу которых относится и миварный подход. Пришло понимание, что системе не обязательно решать особо сложные проблемы без человеческого вмешательства. «Многие современные программы доказательств работают как умные помощники, предоставляя людям разбивать задачи на подзадачи и продумывать эвристики для перебора в пространстве возможных обоснований» [33, с. 44]. Этот вывод может служить обоснованием нашего утверждения, что ИИ – это усилитель человеческих способностей и автоматизация мыслительных процессов. Кроме того, это косвенно подтверждает необходимость введения шкалы измерений интеллектуальности автоматических систем и наличие относительно слабых форм интеллекта у уже существующих компьютерных программ и устройств. Значит, в таком смысле ИИ уже существует и продолжает развиваться, помогая человеку решать сложные задачи. Миварный подход позволяет работать с

различными формами представления данных и правил (знаний), включая и работу с бесконечными описаниями сущностей – вещей, отношений и свойств в миварном многомерном динамическом информационном пространстве унифицированного представления данных и правил (знаний). Если учитывать, что в области ИИ существуют две фундаментальные проблемы – представление знаний и поиск, то миварные сети позволяют по-новому представлять информацию, что, в свою очередь, позволило кардинально ускорить поиск решений и перейти от NP-сложности логического вывода к линейной вычислительной сложности миварного метода обработки данных и правил.

Представление продукций в виде двудольных графов

Перейдем к формализованному представлению продукций и сетей, которые могут быть сформированы из них. Выше было отмечено, что Д.А. Поспелов [24, с. 83-84] представляет сеть правил в таком виде, что она отображается в виде однодольного графа. В работах [1-22] предложен подход по взаимосвязи сетей правил и графов. Более того, показано, что некоторые задачи логического вывода можно решать на основе подходов теории графов. Сети правил и процедур целесообразнее представлять в виде двудольных графов, получая нечто, аналогичное сетям Петри, но с соответствующим развитием до миварных логических сетей [1-22]. Итак, напомним определения двудольных графов, которые необходимы для продолжения анализа. «Граф $G = (V, E)$ называется двудольным, если существует разбиение $V = \{V1, V2\}$, такое, что никакие две вершины из $V1$ или из $V2$ не являются смежными» [26, с. 223]. «Двудольным графом $G = (X, Y, E)$ называется неориентированный граф, вершины которого можно разбить на два класса X и Y так, что концы каждого ребра принадлежат разным классам» [25, с. 125]. Приведем следующее важное замечание: «Введенные понятия допускают естественное обобщение. Неориентированный граф называется k-дольным, если его вершины можно разбить на k классов так, что концы каждого ребра принадлежат разным классам» [25, с. 125]. Таким образом, можно использовать и двудольные, и трехдольные, и многодольные (k-дольные) графы для разных предметных областей. Развивается новый подход к единому представлению знаний и данных, который получил название «миварный». Миварный подход является обобщением и развитием продукционного подхода, сетей Петри и других формализмов, применяемых для логической обработки данных.

Миварные сети

Миварные сети могут быть представлены в виде двудольного графа, состоящего из объектов-переменных и правил-процедур. Для этого, прежде всего, составляются два списка, которые и образуют две непересекающиеся доли графа: список объектов и список правил.

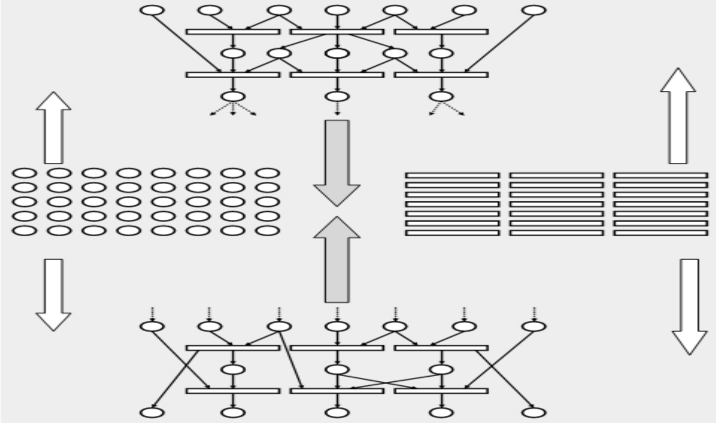


Рисунок 2 – Представление миварной сети в виде двудольного графа

Объекты изображены овалами (кружочками) на рис. 2. Каждое правило в миварной сети является развитием продукций, гиперправил с мультиактивизаторами или вычислительных процедур. Доказано, что с точки зрения дальнейшей их обработки все эти формализмы идентичны и представляют собой, по сути, вершины двудольного графа, которые изображены прямоугольниками на рис. 2.

На рис. 3 показан пример записи «объектов» и «правил» миварной сети в формате XML, а алгоритм логико-вычислительной обработки для программного комплекса УДАВ показан на рис. 4 [10-22].

```
<?xml version="1.0" encoding="UTF-8" ?>
-<root>
-<parametrs>
<parametr id="P1" value="0.0" description="Угол A, против. стороне a (град)" />
<parametr id="P2" value="0.0" description="Угол B, против. стороне b (град)" />
<parametr id="P3" value="0.0" description="Угол C, против. Стороне c (град)" />
...
</parametrs>
-<rules>
<rule id="R1" resultId="P1" initId="P2,P3" value="180-P2-P3" description="Сумма" />
<rule id="R2" resultId="P2" initId="P1,P3" value="180-P1-P3" description="Сумма" />
<rule id="R3" resultId="P3" initId="P1,P2" value="180-P1-P2" description="Сумма" />
...
</rules>
-<metadata>
<idParametr inc="33" />
<idRule inc="161" />
</metadata>
</root>
```

Рисунок 3 – Пример записи объектов и правил в формате XML

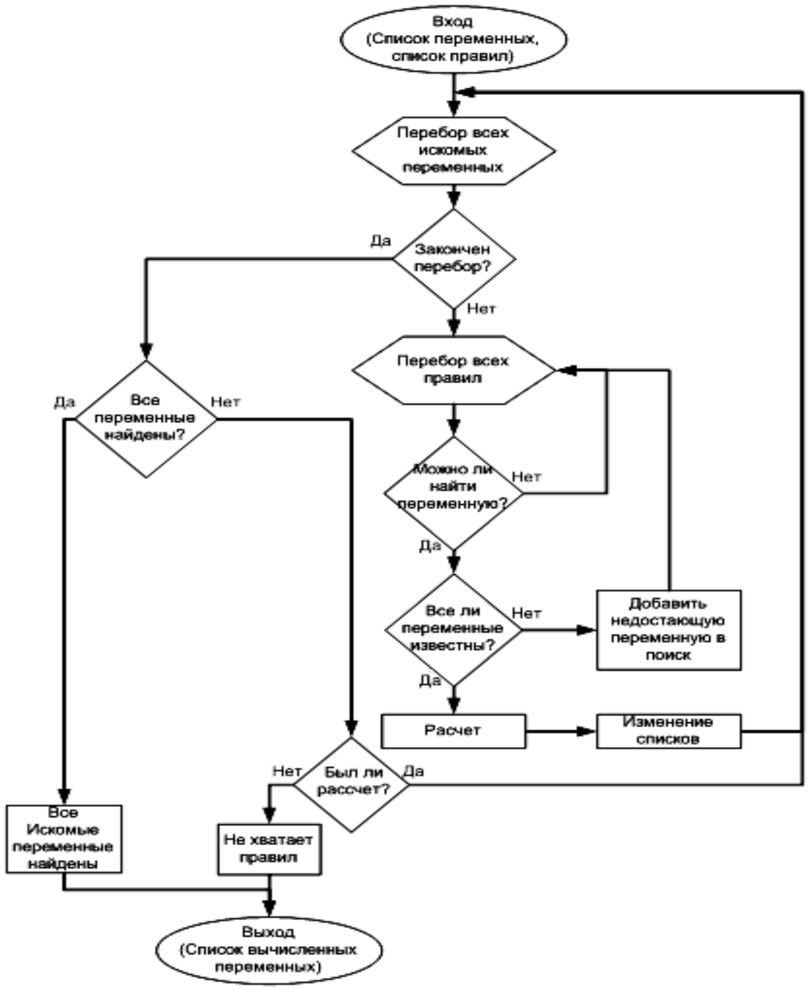


Рисунок 4 – Алгоритм миварной логико-вычислительной обработки для ПК УДАВ

Миварный метод логико-вычислительной обработки

Кратко опишем теоретические основы работы миварного метода логико-вычислительной обработки данных [2-4], [10], [15,] [20]. Суть работы метода: для миварной сети логических правил, представленной в списочной форме, строится матрица. Затем, на основе анализа этой матрицы определяется факт наличия маршрута вывода, потом определяются возможные маршруты вывода, а на последнем этапе из этих маршрутов выбирают «кратчайший», наиболее оптимальный по заданным критериям. Прежде всего, надо создать формализованное описание предметной области. Для этого выделяют объекты-переменные и правила-процедуры, а затем создают соответствующие списки «объектов» и «правил». Формализованное представление метода логико-вычислительной обработки данных аналогично двудольному графу миварной логической сети, показанному на рис. 2. Выделим **три основных этапа** миварной обработки данных:

1) формирование миварной матрицы описания предметной области;

2) работа с матрицей и конструирование алгоритма решения заданной задачи;

3) по полученному алгоритму выполнение всех вычислений и нахождение ответа.

Первый этап по сути является этапом синтеза концептуальной модели предметной области и ее формализации в виде продукционных правил с переходом на миварные правила: ***«Входные объекты – правила/процедуры – выходные объекты»***. В настоящее время этот этап является наиболее сложным и требует участия человека (эксперта) для создания миварной модели предметной области.

На втором этапе собственно и выполняется автоматическое конструирование алгоритма решения или логический вывод. Исходными данными для этого являются – миварная матрица описания предметной области, а также заданные входные («ДАНО») и искомые («НАЙТИ») объекты-переменные.

На третьем этапе должно выполняться решение по полученному алгоритму. Однако, в настоящее время в программном комплексе УДАВ совмещены работы по второму и третьему этапам. Отметим, что уже создано более 7 различных реализаций миварного метода. Часть этих реализаций в виде программных комплексов выделяет все три основных этапа и обрабатывает их отдельно. В данной работе основное внимание уделим программному комплексу УДАВ, в котором все три этапа могут быть совмещены.

Пусть известны ***m* правил** и ***n* переменных**. Тогда в матрице $V\ (m \times n)$, каждая строка которой соответствует одному из правил и содержит информацию об используемых в правиле переменных, могут быть представлены все взаимосвязи между правилами и переменными. При этом в каждой строке все входные переменные этого правила на соответствующих позициях матрицы помечаются символом x, все выходные – y; все переменные, которые уже получили в процессе вывода или задания исходных данных некоторое конкретное значение, – z; а все искомые (выходные) переменные, т.е. те, которые необходимо «вывести» из исходных (входных) данных, – w. Кроме того, добавим в матрицу V одну строку и один столбец для хранения в них служебной информации. Напомним, что количество служебной информации может видоизменяться для различных конкретных реализаций программных комплексов. Получаем матрицу V размерности $(m+1) \times (n+1)$, в которой отражена вся структура исходной сети правил. Сразу отметим, что структура этой логической сети может изменяться в любое время, т.е. это сеть правил с изменяемой (эволюционной) структурой. Пример такой матрицы показан на рис. 5. На рис. 6 эта же матрица показана в виде исходного задания двудольного графа миварной логической сети. Для удобства читателей все рисунки собраны вместе и показаны подряд, после их описания. Опишем **пример работы метода.**

Для поиска маршрута логического вывода производят действия по следующим этапам.

1. В строке *(m+1)* известные переменные помечают *z*, а искомые – *w*. Например, на рис. 7 в строке *(m+1)* символом z отмечены позиции: 1, 2, 3, а символом *w* – позиция *(n-2)*. На рис. 8 это действие показано на примере построения двудольного графа миварной сети.

2. Осуществляют последовательно, например, сверху вниз, поиск таких правил, которые могут быть активизированы, т.е. у которых известны все входные переменные. Если таких правил нет, то маршрута логического вывода нет и необходимо выдать запрос на уточнение (добавление) входных данных. Если такие правила есть, то у каждого из них в служебной строке делается пометка. Например, можно ставить в этих клетках таблицы цифру 1, что и показано на рис. 9 в клетке *(1, n+1)*. На рис. 10 это действие показано на примере построения двудольного графа.

3. Если таких правил несколько, то осуществляется выбор по заранее определенным критериям такого или таких правил, которые должны быть активизированы в первую очередь. При наличии достаточных ресурсов, одновременно могут запускаться сразу несколько правил.

4. Имитация запуска правила (процедуры) осуществляется путем присваивания выводимым в этом правиле переменным значений «известно», т.е., в нашем примере, – *z*. Запущенное правило, для удобства дальнейшей работы, помечается дополнительно, например, цифрой 2 (это не обязательно). Например, как показано на рис. 11: в клетках *(m+1, n-1)*, *(m+1, n)* и *(1, n+1)* произведены соответствующие изменения. На рис. 12 это показано в виде миварной сети.

5. После имитации запуска правил проводят анализ достижения цели, т.е. анализируют получение требуемых значений путем сравнения служебных символов в служебной строке. Если в служебной строке *(m+1)* осталось хоть одно значение «искомая» (т.е. *w*), то осуществляют дальнейший поиск маршрута логического вывода. В противном случае задача считается успешно решенной, а все задействованные правила в соответствующем порядке их запуска образуют искомый маршрут логического вывода. На рис. 11 и 12 запущено только одно правило, но искомые значения не определены, значит надо переходить к следующему этапу.

6. Прежде всего, определяют наличие таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе. Если таких правил нет, то маршрута вывода нет и поступают аналогично пункту 2 этого метода. Если такие правила есть, то продолжают поиск маршрута вывода. В нашем примере такие правила есть, что и показано на рис. 13: в клетке *(2, n+1)* появилась 1 как признак возможности запуска этого правила. На рис. 14 это действие показано на примере построения двудольного графа миварной логической сети.

7. На следующем этапе опять, аналогично этапу 4, запускают правила, далее, аналогично этапам 5 и 6, выполняют необходимые действия столько раз, сколько требуется для получения результата. При необходимости повторяют все этапы со 2 по 7 до достижения результата. При этом результат может быть как положительный – маршрут вывода существует, так и отрицательный – вывода нет из-за неопределенности входных данных. Для наглядности продолжим пошаговое выполнение нашего примера. Итак, необходимо провести имитацию запуска правила 2, что и показано на рис. 15. На рис. 16 это действие показано на миварной сети.

8. В клетках *(m+1, 4)* и *(m+1, 5)* получаем признак выводимости переменных 4 и 5, а в клетке *(2, n+1)* формируем признак того, что правило уже было запущено, т.е. ставим цифру 2. После этого проводим анализ служебной строки и видим, что не

все искомые переменные известны. Значит необходимо продолжить обработку матрицы *V* размерности *(m+1)* × *(n+1)*. Анализ этой матрицы показывает возможность запуска правила *m*, что и отражено на рис. 17, а на рис. 18 это же действие показано на примере построения двудольного графа миварной сети.

9. Продолжим дальнейшую обработку примера. При запуске правила m появляются новые значения, в том числе и для искомых переменных, что и показано на рис. 19. На рис. 20 это же действие показано на примере построения двудольного графа миварной логической сети.

10. Итак, в нашем примере в служебной строке больше не осталось искомых правил, а в клетках таблицы появились новые значения: в клетке *(m, n+1)* – 2, а в клетке *(m+1, n–2)* вместо значения w появилось значение *z*. Таким образом, получен положительный результат, следовательно, маршрут логического вывода при данных исходных значениях существует.

Теперь приведем рис. 5 – 20, которые подробно описаны выше.

V	1	2	3	4	5	…	n-2	n-1	n	n+1
1	X	X	X					Y	Y	
2			X	Y	Y			X	X	
…						…				
m		X		X	X		Y			
m+1										

Рисунок 5 – Пример исходного задания матрицы V размерности *(m+1)* × *(n+1)*

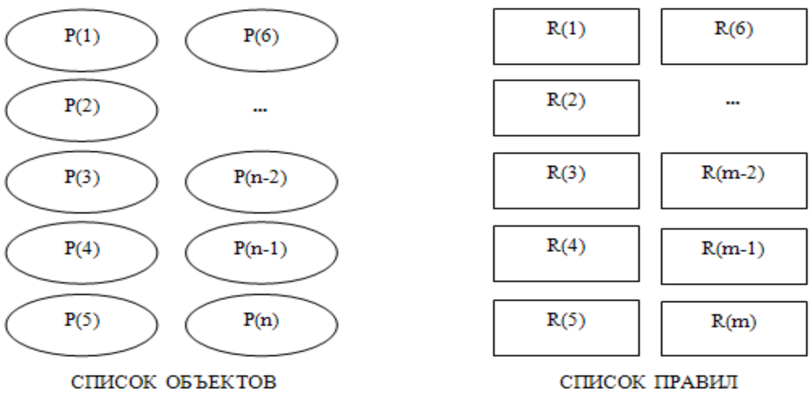


Рисунок 6 – Пример исходного задания двудольного графа миварной логической сети

V	1	2	3	4	5	…	n-2	n-1	n	n+1
1	X	X	X					Y	Y	
2			X	Y	Y			X	X	
…						…				
m		X		X	X		Y			
m+1	Z	Z	Z				W			

Рисунок 7 – Шаг № 1 примера: постановка задачи «дано» и «найти»

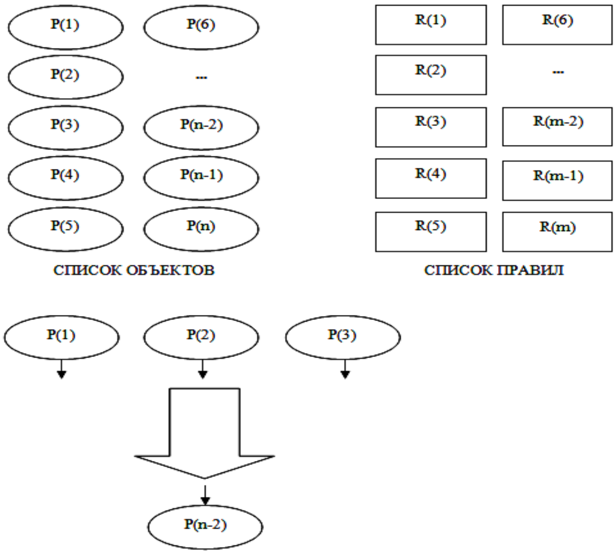


Рисунок 8 – Иллюстрация при помощи двудольного графа шага № 1

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	1
2			X	Y	Y			X	X	
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z				W			

Рисунок 9 – Шаг № 2 примера обработки матрицы *V*

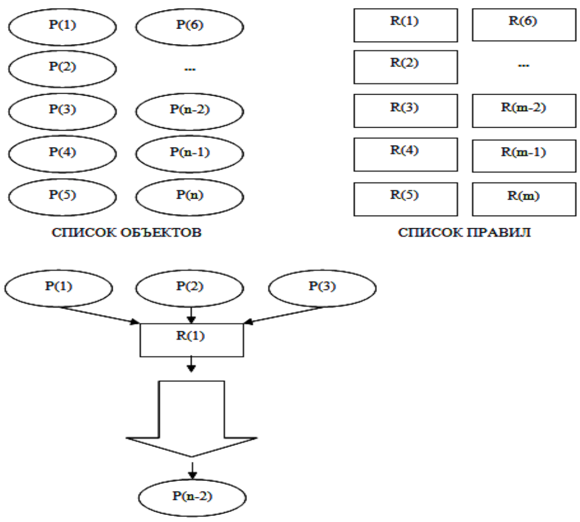


Рисунок 10 – Иллюстрация при помощи двудольного графа шага № 2

V	**1**	**2**	**3**	**4**	**5**	**...**	**n-2**	**n-1**	**n**	**n+1**
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z				W	Z	Z	

Рисунок 11 – Шаг № 3 примера обработки матрицы *V*

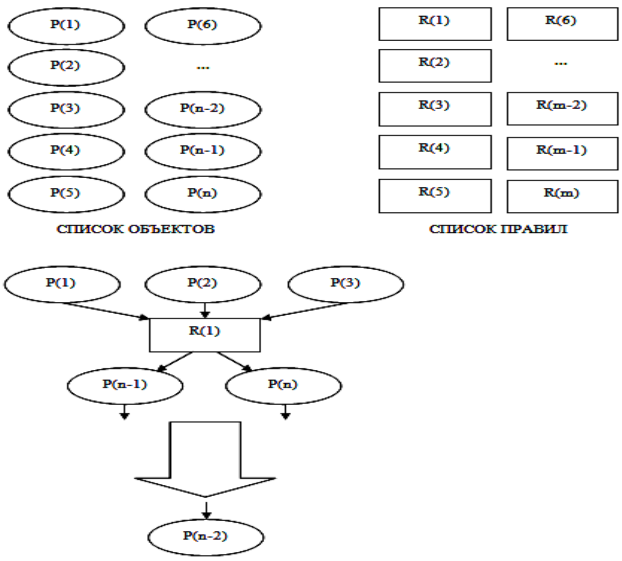


Рисунок 12 – Иллюстрация при помощи двудольного графа шага № 3

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	1
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z				W	Z	Z	

Рисунок 13 – Шаг № 4 примера обработки матрицы *V*

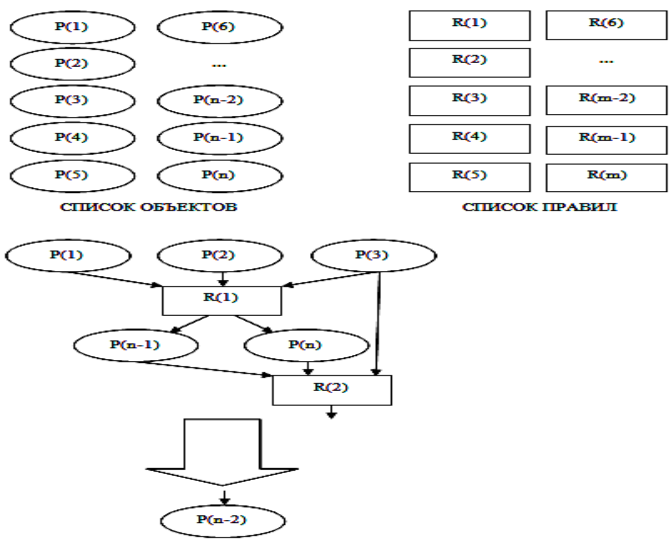


Рисунок 14 – Иллюстрация при помощи двудольного графа шага № 4

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	2
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z	Z	Z		W	Z	Z	

Рисунок 15 – Шаг № 5 примера обработки матрицы *V*

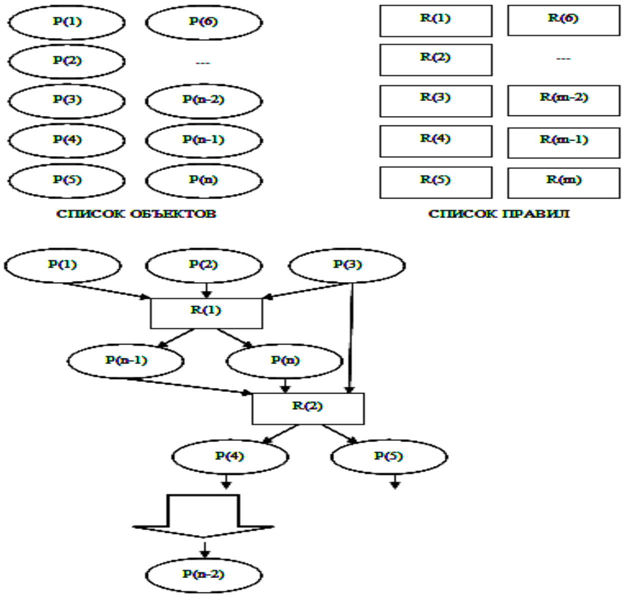


Рисунок 16 – Иллюстрация при помощи двудольного графа шага № 5

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	2
...						...				
m		X		X	X		Y			1
m+1	Z	Z	Z	Z	Z		W	Z	Z	

Рисунок 17 – Шаг № 6 примера обработки матрицы *V*

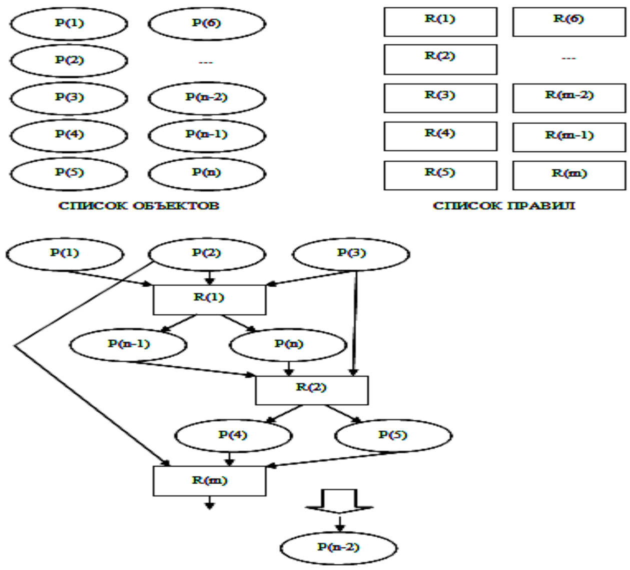


Рисунок 18 – Иллюстрация при помощи двудольного графа шага № 6

V	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	2
...						...				
m		X		X	X		Y			2
m+1	Z	Z	Z	Z	Z		Z(W)	Z	Z	

Рисунок 19 – Шаг № 7 примера обработки матрицы *V*

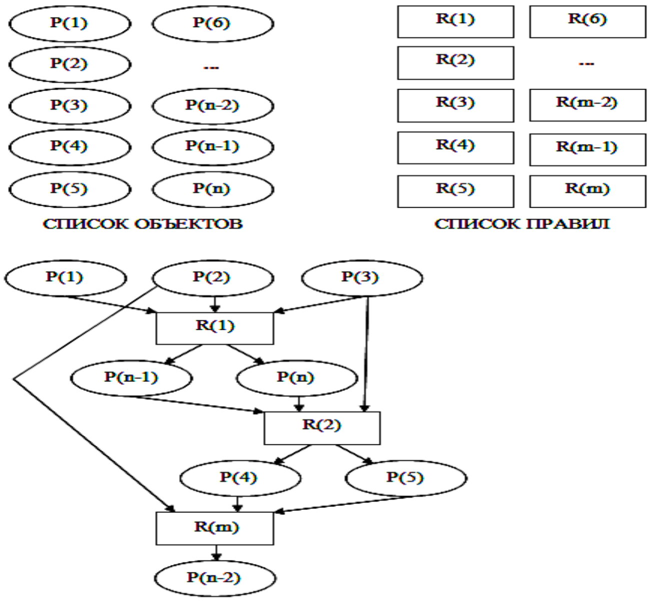


Рисунок 20 – Иллюстрация при помощи двудольного графа шага № 7

Преимущества и новизна миварного метода обработки

Таким образом, показано, что для экспертных систем на основе продукционного подхода и вместо традиционных однодольных графов необходимо и достаточно использовать двудольные графы миварных сетей. С научно-практической точки зрения основную сложность при использовании экспертных систем составляет именно концептуальное, продукционное описание предметной области и формирование необходимых двух списков: объектов и правил для миварных логических сетей. Непосредственная обработка проводится на основе универсального механизма, описанного в [1-23]. Выделенные правила-процедуры при необходимости могут быть представлены в виде продукций, что соответствует традиционному подходу, но в то же время позволяет реализовать многомерное эволюционное представление данных. Универсальные возможности миварного подхода обусловлены тем, что он обобщает все известные модели данных, включая семантические сети, модель «сущность-связь» и онтологии [4], [10], [15], [18-20].

Основными **преимуществами миварного подхода** являются:

1) линейная вычислительная сложность и реальное время работы;

2) решение логических и вычислительных (и других) задач;

3) управление потоком входных данных и оперативная диагностика;

4) адаптивное описание и непрерывное решение задач;

5)активная работа с запросами или уточнениями входных данных на эволюционной сети правил и объектов (самообучение).

Теоретический расчет вычислительной сложности миварного метода обработки

Общее количество действий миварного метода определяется суммой действий на каждом этапе:

1) присваивание известных z и искомых w значений клеткам служебной строки $(m+1)$ (общее количество таких действий не более n);

2) присвоение признака обработки правил в служебном столбце $(n+1)$ (количество действий не более $2m$, но может быть не более числа m);

3) присваивание признака известности (*z*) клеткам служебной строки *(m+1)* (общее количество таких действий не более n);

4) определение новых значений клеткам строки (m+1) (количество таких действий не более *n*).

Действия на этапах 1, 2 и 3 выполняются над одним массивом данных – строкой *(m+1)*. При этом общее, суммарное, количество действий на всех этих этапах (1, 2 и 3) не должно превышать общего количества клеток в этой строке, так как обработанные значения «вычеркиваются» и более не обрабатываются. Получаем, что общее количество действий (KD) при миварном методе обработки, т.е. его вычислительная сложность, не превышает количества клеток в служебных частях матрицы: ***O(m+n)***, т.е. $KD \leq (m+n)$.

В худшем случае по каждой клетке матрицы «проходят» один раз. Тогда миварный метод решает задачу поиска логического вывода с вычислительной сложностью: $KD = O(m \cdot n)$, (примечание: «***m*** умножить на ***n***»), т.е. **имеет место линейная зависимость** либо от количества правил, либо от количества переменных. Таким образом, миварный метод обладает линейной вычислительной сложностью обработки.

Результаты экспериментов

В настоящее время создано уже несколько программных комплексов – прототипов экспертных систем, которые реализовали на практике миварный метод обработки данных и правил. Прежде всего, это программный комплекс УДАВ [9-22], на основе которого начата реализация проекта «Миварная активная энциклопедия». В Миварной энциклопедии описано три предметные области: геометрия – «треугольники», «стереометрия» и «окружности». Г.С. Сергушиным реализован программный комплекс «физика падающего тела» [10], [18], [19] и прототип миварной экспертной системы управления двигателем автомобиля. Коллективом авторов создан прототип экспертной системы – тренажера логических переключений электрических подстанций и распределенных электросетей с трехмерным интерфейсом. Этот тренажер позволяет адаптивно перестраивать конфигурацию логической схемы и самостоятельно добавлять пользователям новые элементы.

Как было указано выше, наибольшую трудность составляет описание предметной области в виде миварной сети. Поэтому для проведения вычислительных экспериментов был создан специальный программный комплекс, основанный на программе УДАВ. Прежде всего, в новом тестовом комплексе реализован генератор миварных матриц описания предметных областей. Такое решение было необходимо для создания миварных матриц больших размеров. Например, в процессе экспериментов были созданы миварные матрицы размерности 1 170 007 объектов на 3 510 015 правил, т.е. матрица миллион на 3,5 миллиона клеток. Для генерации таких матриц используется некоторый специально разработанный алгоритм, который по некоторым правилам формирует элементарные строки миварной матрицы. На вход этой программе задается количество объектов – переменных. Далее генератор матриц сам генерирует простые арифметические правила для введенного количества объектов. В качестве примера простых правил могут использоваться следующие: ***a+b=c* и/или *c-b=a* и/или *c-a=b***.

Генератор матриц выполняет формирование матрицы и сохраняет ее в специальный файл. Затем тестовая программа считывает тестовую матрицу и начинает обработку. Для однозначности и адекватности проводимых сравнений все генерируемые матрицы в определенном смысле являются подобными. Это обеспечивается за счет алгоритма формирования простых правил и заданием следующих управляющих

данных. В качестве известных объектов всегда берутся объекты P1=10 и P2=10. А в качестве искомого объекта всегда задается последний по номеру объект. Таким образом, если количество входных параметров = 100, то программа будет искать P100, а если количество входных параметров = 10000, то программа будет искать P10000. При необходимости, пользователь может сгенерировать свои матрицы или изменить уже созданные. Такие эксперименты были проведены. Данная тестовая программа написана только для оценки времени работы ядра УДАВ при элементарных правилах и с пропорциональным наращиванием миварных матриц. В конце своей работы тестовая программа строит граф решения, где: розовым цветом показаны параметры и внутри пишется его идентификатор, а оранжевым цветом показано правило и внутри него тоже пишется идентификатор. Соответственно наверху графа рисуются исходные данные, а в внизу – параметр, который искали. Генератор устроен таким образом, чтобы решение всегда существовало, а его нахождение требовало пропорционального увеличения числа шагов искомого алгоритма решения задачи.

Отметим, что время работы программы определяется по внутренним часам компьютера и может сильно зависеть от общей нагрузки и параллельно выполняемых задач. В настоящий момент реализованы и однопоточный вариант программы, и многопоточный. Но большая часть экспериментов была проведена на однопоточном варианте, которому и уделено основное внимание. Подчеркнем, что тестовая программа не учитывает время создания тестовой матрицы на первом этапе и время формирования графа решения в конце. Время решения задач выводится в милисекундах (ms). Учитывая сформулированные особенности работы тестовой программы, всем испытателям было рекомендовано запускать тестовую программу с определенным шагом увеличения объектов и в относительно равнозначных условиях внешней загрузки компьютера. Например, было рекомендовано отключить все внешние функции и не запускать параллельно никаких программ. В процессе проведения экспериментов выявили, что основным ограничением по количеству обрабатываемых объектов является ресурс оперативной памяти компьютера и корректность работы с ней операционной системы.

Для проведения тестов были задействованы различные компьютеры, начиная от простейших нетбуков с оперативной памятью 512 Мб и до небольших серверов с процессорами Intel 3,8 Ггц и оперативной памятью 4 Гб. Количество ядер не влияло на скорость работы, т.к. основные тесты проводились с использованием однопоточной программы. Использовались два семейства операционных систем: Windows и MacOS. В совокупности, эксперименты были проведены более чем на 15 компьютерах. Для каждого значения количества объектов-переменных проводилось 10 однотипных испытаний. Расчеты показали, что ошибка измерения времени работы программы не превышала 3%. Все проведенные эксперименты подтвердили на практике линейную вычислительную сложность миварного метода обработки данных и правил. Следовательно, проведенные вычислительные эксперименты подтверждают теоретические выводы о линейной вычислительной сложности миварного метода обработки.

Отметим, что на семействе операционных систем Windows удалось получить следующие результаты – до 150 000 объектов и 450 000 правил на компьютерах с оперативной памятью 4 Гб. В зависимости от процессора, время решения составляло от нескольких миллисекунд до 2 793 672 миллисекунд, что примерно составляет 47 минут.

Наибольший интерес представляют эксперименты, проведенные на ноутбуке MacBook, операционная система MacOSX версии 10.6.7, процессор Intel Core 2DUO с частотой 2 Ггц и с оперативной памятью DDR3 объемом 4 Гб. На этом ноутбуке в различные периоды времени были проведены тестовые испытания двух версий тестовой программы: B1 и B2. Результаты испытаний приведены на рис. 21. Итак, на

обычном серийном ноутбуке MacBook тестовая программа, написанная на языке ЯВА (не самом быстром языке), движок УДАВ обрабатывает миварные матрицы размерности миллион объектов на 3,5 миллиона параметров за время, примерно равное 3 часам.

Испытания на максимальное значение получили следующий результат: 1 170 007 объектов – переменных, 3 510 015 правил и время решения 12 239 183 миллисекунды, что примерно равно 200 минутам и немногим более 3 часов. 3,5 миллиона правил – это очень большое число. Для сравнения приведем следующие оценки: система управления крупными атомными станциями оперирует всего 20 000 правил; описание всей программы средней школы и технического ВУЗа потребует около 300 правил на каждую научную область и в сумме не превысит 100 000 правил. Напомним, что классические продукционные и предикатные логические системы не могли работать и с сотней правил. Таким образом, возможности программного комплекса УДАВ и миварного метода обработки позволяют создать глобальную экспертную систему, в которой будут собраны все формализованные знания человечества. Этот проект мы называем: «Миварная активная энциклопедия».

Если у кого-то еще остаются сомнения, то предлагается создать тестовый пример, разработать тестовые миварные матрицы и прислать их для проведения эксперимента. Будет выполнена загрузка матриц в миварный движок УДАВ, получен ответ и распечатан граф алгоритма решения тестовой задачи. При необходимости, можно провести совместные эксперименты, т.к. уже проведенные исследования достаточно полно позволили проверить и протестировать миварный движок УДАВ.

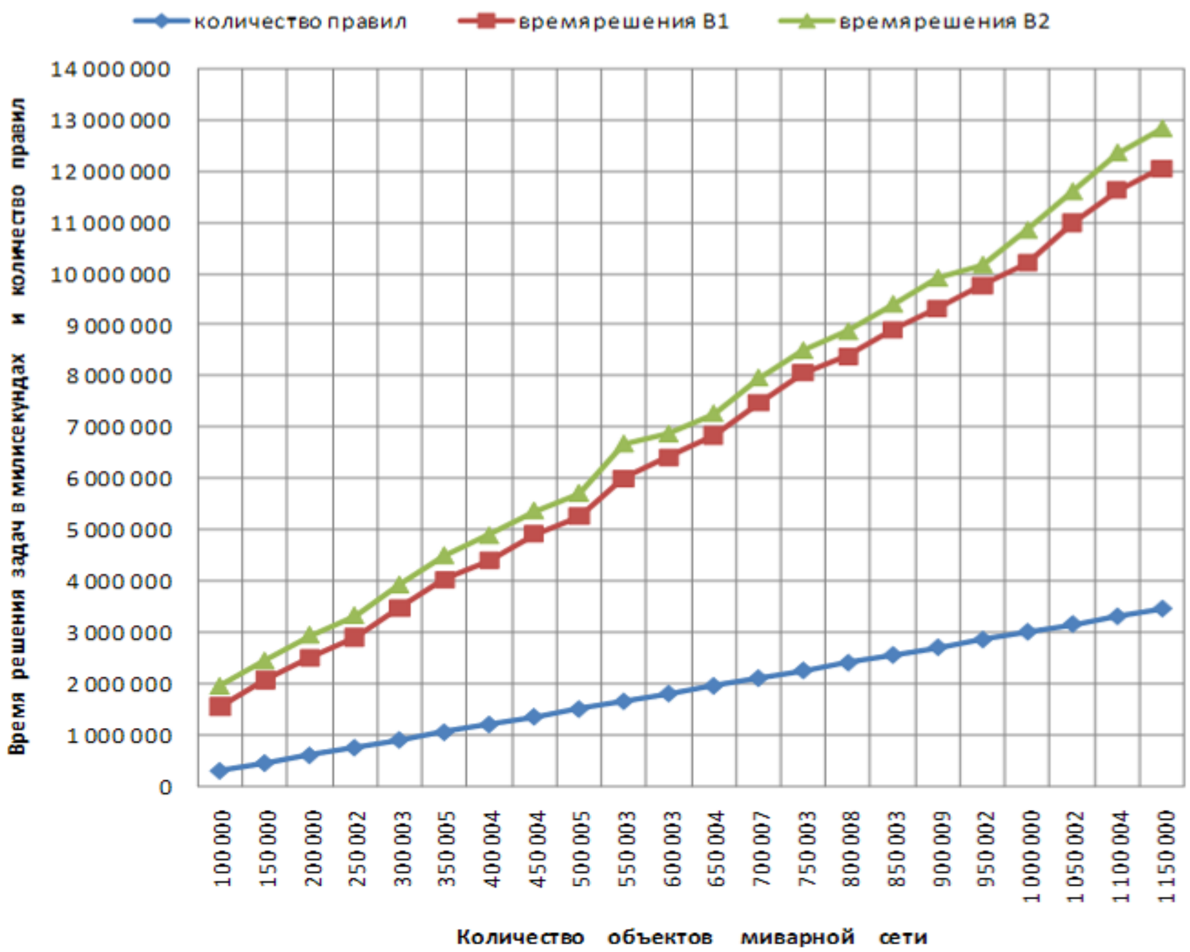


Рисунок 21 – Результаты практических исследований и экспериментов

На рис. 22 приведена иллюстрация перехода от однодольных продукционных систем к двудольным миварным сетям.

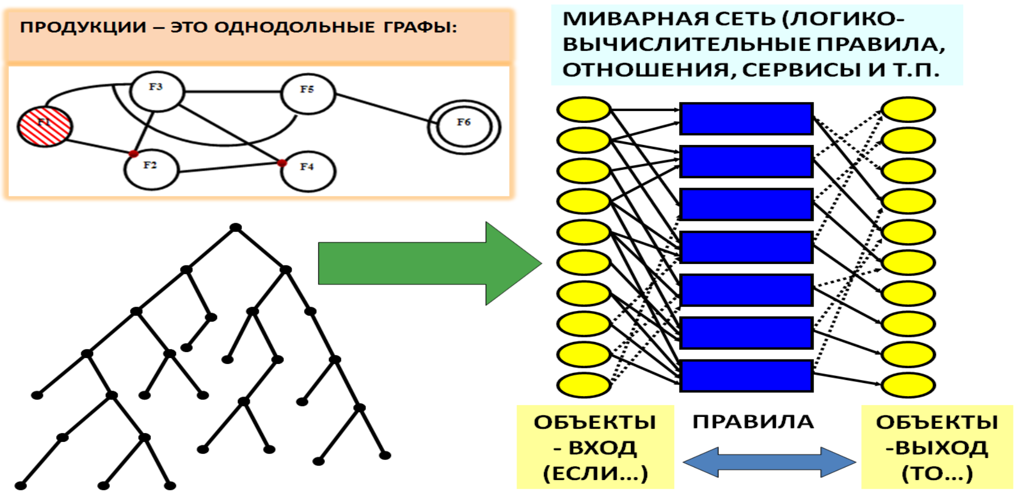


Рисунок 22 – Переход от продукций к миварным сетям

Выводы

Показан теоретический переход от однодольных продукционных систем к двудольным миварным логико-вычислительным сетям. Приведены примеры реализации миварных сетей в формализмах матриц и графов. Теоретически обоснована линейная вычислительная сложность логического вывода и/или автоматического конструирования алгоритмов из переменных-объектов и правил-процедур миварных сетей.

В качестве миварных правил могут быть использованы различные сервисы, модули, вычислительные процедуры и т.п. Автоматический конструктор алгоритмов может использоваться для поиска логического вывода в области создания экспертных систем и интеллектуальных пакетов прикладных программ (решателей задач).

На основе миварных сетей создан программный комплекс УДАВ, который **обрабатывает** более 1,17 млн переменных и **более 3,5 млн правил** на обычных компьютерах и ноутбуках. Приведены результаты практических расчетов и решений различных прикладных задач, которые на практике подтверждают линейную вычислительную сложность конструирования алгоритмов в формализме миварных сетей.

Программный комплекс УДАВ используется как для решения логических, так и вычислительных задач. Приведены сведения о практической реализации нескольких миварных экспертных систем. Миварные сети позволяют перейти к новому поколению экспертных систем и интеллектуальных пакетов прикладных программ.

Подробно описан линейной вычислительной сложности миварный метод логико-вычислительной обработки данных и правил. Обосновано, что для экспертных систем нового поколения вместо продукционного подхода и традиционных однодольных графов целесообразно использовать двудольные графы миварных сетей.

Практическая реализация миварного метода в программном комплексе УДАВ и проведение вычислительных экспериментов на различных компьютерах убедительно доказывает линейную вычислительную сложность миварного метода. Впервые предоставлены возможности создания мультипредметных экспертных систем, способных на практике работать с миллионами объектов и правил миварных сетей.

Миварные технологии накопления и обработки информации значительно расширяют технические возможности компьютеров и позволяют создавать принципиально

новые экспертные системы, интеллектуальные пакеты прикладных программ и интеллектуальные системы. На основе миваров могут быть построены активные, обучаемые, логически рассуждающие, автоматические системы, которые будут служить прообразом систем искусственного интеллекта.

Литература

1. Варламов О.О. Алгоритм разреза сети по вершинам и ребрам ее графа сложности O(n^2) / О.О Варламов // Труды НИИР : сб. ст. – М., 1997. – 136 с. – С. 92-97.
2. Варламов О.О. Разработка адаптивного механизма логического вывода на эволюционной интерактивной сети гиперправил с мультиактивизаторами, управляемой потоком данных / О.О. Варламов // Искусственный интеллект. – 2002. – № 3. – С. 363-370.
3. Варламов О.О. Разработка линейного матричного метода определения маршрута логического вывода на адаптивной сети правил / О.О. Варламов // Известия вузов. Электроника. – 2002. – № 6. – С. 43-51.
4. Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство / Варламов О.О. – М. : Радио и связь, 2002. – 288 с.
5. Варламов О.О. Основы многомерного информационного развивающегося (миварного) пространства представления данных и правил / О.О. Варламов // Информационные технологии. – 2003. – № 5. – С. 42-47.
6. Варламов О.О. Системный анализ и синтез моделей данных и методы обработки информации в самоорганизующихся комплексах оперативной диагностики : дисс. … на соискание ученой степени доктора технических наук / О.О. Варламов. – М. : МАРТИТ, 2003. – 307 с.
7. Варламов О.О. Системный анализ и синтез моделей данных и методы обработки информации в самоорганизующихся комплексах оперативной диагностики : автореф. дисс. на соискание ученой степени доктора технических наук / О.О. Варламов. – М. : МАРТИТ, 2003. – 44 с.
8. Варламов О.О. Эволюционные базы данных и знаний. Миварное информационное пространство / О.О. Варламов // Известия Южного федерального университета. Технические науки. – 2007. – Т. 77, № 2. – С. 77-81.
9. Владимиров А.Н., Применение многопроцессорного вычислительного кластера НИИР для распараллеливания алгоритмов в научно-технических и вычислительных задачах / А.Н. Владимиров, О.О. Варламов, А.В. Носов, Т.С. Потапова // Труды НИИ радио. 2009. – № 3. – С. 120-123.
10. Санду Р.А. Миварный подход к созданию интеллектуальных систем и искусственного интеллекта. Результаты 25 лет развития и ближайшие перспективы / Р.А. Санду, О.О. Варламов. – М. : Стандартинформ, 2010. – 339 с.
11. Развитие миварного метода логико-вычислительной обработки информации для АСУ, тренажеров, экспертных систем реального времени и архитектур, ориентированных на сервисы / О.О. Варламов, Р.А. Санду, А.Н. Владимиров [и др.] // Труды НИИР. – 2010. – .№ 3. – С. 18-26.
12. Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил / А.Н. Владимиров, О.О. Варламов, А.В. Носов // Труды НИИ радио. – 2010. – Т. 1. – С. 108-116.
13. Варламов О.О. Миварный подход к разработке интеллектуальных систем и проект создания мультипредметной активной миварной интернет-энциклопедии / О.О. Варламов // Известия Кабардино-Балкарского научного центра РАН. – 2011. – № 1. – С. 55-64.
14. Подкосова Я.Г. Использование технологий виртуальной реальности для трехмерной визуализации результатов моделирования и для миварных обучающих систем / Я.Г. Подкосова, С.А. Васюгова, О.О. Варламов // Известия Кабардино-Балкарского научного центра РАН. – 2011. – № 1. – С. 226-232.
15. Варламов О.О. Обзор 25 лет развития миварного подхода к разработке интеллектуальных систем и создания искусственного интеллекта / О.О. Варламов // Труды НИИ радио. – 2011. – № 1. – С. 34-44.
16. Анализ перспектив использования технологий виртуальной реальности в дистанционном обучении / Я.Г. Подкосова, О.О. Варламов, А.В. Остроух [и др.] // Вопросы современной науки и практики. Университет им. В.И. Вернадского. – 2011. – № 2 (33). – С. 104-111.
17. Подкосова Я.Г. Новые возможности и ограничения технологий виртуальной реальности для проведения научных исследований, трехмерной визуализации результатов моделирования и создания миварных обучающих систем и тренажеров / Я.Г. Подкосова, С.А. Васюгова, О.О. Варламов // Труды Научно-исследовательского института радио. – 2011. – № 2. – С. 5-16.
18. Материалы веб-сайта д.т.н. О.О. Варламов [Электронный ресурс]. – Режим доступа : // http://www.ovar.narod.ru. 2012.

19. Материалы веб-сайта компании МИВАР [Электронный ресурс]. – Режим доступа : // http://www.mivar.ru. 2012.
20. Varlamov O.O. MIVAR technologies of the development of intelligent systems and the creation of the active multi-subject online MIVAR encyclopaedia / O.O. Varlamov // Pattern Recognition and Information Processing (PRIP`2011): proceedings of the 11th International Conference (18 – 20 may, Minsk, Republic of Belarus). – Minsk : BSUIR, 2011. – 472 p. – P. 326-329.
21. Podkosova Y.G. MIVAR learning systems, virtual reality and 3D – visualization of scientific modelling results / Y.G. Podkosova, S.A. Vasiuhova, O.O. Varlamov // Pattern Recognition and Information Processing (PRIP`2011): proceedings of the 11th International Conference (18 – 20 may, Minsk, Republic of Belarus). – Minsk : BSUIR, 2011. – 472 p. – P. 447-450.
22. About the prospects of active Mivar's Internet-encyclopedia / A.V. Nosov, L.E. Adamova, O.O. Varlamov [et al.] // Proceedings of the Congress on intelligent systems and information technologies «AIS-IT'10». Scientific publication in 4 volumes. – Moscow : Physmathlit. – 2010. – Vol. 4. – 128 p. – P. 118-119.
23. Varlamov O.O. MIVAR: Transition from Productions to Bipartite Graphs MIVAR Nets and Practical Realization of Automated Constructor of Algorithms Handling More than Three Million Production Rules / O.O. Varlamov // [Электронный ресурс] 2012. – Режим доступа : http://arxiv.org/abs/1111.1321. Свободный (Дата обращения: 31.05.2012).
24. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов / Поспелов Д.В. – М. : Радио и связь, 1989. – 184 с.
25. Кузнецов О.П. Дискретная математика для инженера. 6-е изд. / Кузнецов О.П. – СПб. : Лань, 2009. – 400 с.
26. Кук Д. Компьютерная математика / Д. Кук, Г. Бейз ; [пер. с англ.]. – М. : Наука, Физ.-мат. лит., 1990. – 384 с.
27. Саймон А.Р. Стратегические технологии баз данных: менеджмент на 2000 год / А.Р. Саймон ; [под ред. и с предисл. М.Р. Когаловского ; пер. с англ.]. – М. : Финансы и статистика, 1999. – 479 с.
28. Марков А.С. Базы данных. Введение в теорию и методологию : учебник / А.С. Марков, К.Ю. Лисовский. – М. : Финансы и статистика, 2006. – 512 с.
29. Кузнецов С.Д. Базы данных: языки и модели / Кузнецов С.Д. – М. : ООО «Бином-Пресс», 2008. – 720 с.
30. Когаловский М.Р. Энциклопедия технологий баз данных / Когаловский М.Р. – М. : Финансы и статистика, 2005. – 800 с.
31. Ахо А.В., Структуры данных и алгоритмы / А.В. Ахо, Д.Э. Хопкрофт, Д.Д. Ульман. – М. : Издательский дом «Вильямс», 2007. – 400 с.
32. Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В.Ф. Хорошевский // Искусственный интеллект и принятие решений. – 2008. – № 1. – С. 80-97.
33. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. / Дж.Ф. Люгер ; [4-е издание ; пер. с англ.]. – М. : Издательский дом «Вильямс», 2005. – 864 с.

Literatura

1. Varlamov O.O. Trudy NIIR: Sb. st. M. 1997. S. 92-97.
2. Varlamov O.O. Iskusstvennyj intellekt. 2002. № 3. S. 363-370.
3. Varlamov O.O. Izvestija vuzov. Jelektronika. 2002. № 6. S.43-51.
4. Varlamov O.O. Jevoljucionnye bazy danyh i znanij dlja adaptivnogo sinteza intellektual'nyh sistem. Mivarnoe informacionnoe prostranstvo. M.: Radio i svjaz'. 2002. 288 S.
5. Varlamov O.O. Informacionnye tehnologii. 2003. № 5. S. 42-47.
6. Varlamov O.O. Sistemnyj analiz i sintez modelej dannyh i metody obrabotki informacii v samoorganizujushhihsja kompleksah operativnoj diagnostiki: Dissertacija na soiskanie uchenoj stepeni doktora tehnicheskih nauk. M.: MARTIT. 2003. 307 s.
7. Varlamov O.O. Sistemnyj analiz i sintez modelej dannyh i metody obrabotki informacii v samoorganizujushhihsja kompleksah operativnoj diagnostiki: Avtoreferat dissertacii na soiskanie uchenoj stepeni doktora tehnicheskih nauk. M.: MARTIT. 2003. 44 s.
8. Varlamov O.O. Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki. 2007. T. 77. № 2. S. 77-81.
9. Vladimirov A.N. Trudy NII radio. 2009. № 3. S. 120-123.
10. Sandu R.A. Mivarnyj podhod k sozdaniju intellektual'nyh sistem i iskusstvennogo intellekta. Rezul'taty 25 let razvitija i blizhajshie perspektivy. M.: Standartinform. 2010. 339 s.

11. Varlamov O.O. Trudy NIIR. 2010. № 3.S.18-26.
12. Vladimirov A.N. Trudy NII radio. 2010. T. 1. S. 108-116.
13. Varlamov O.O. Izvestija Kabardino-Balkarskogo nauchnogo centra RAN. 2011. № 1. S. 55-64.
14. Podkosova Ja.G. Izvestija Kabardino-Balkarskogo nauchnogo centra RAN.2011.№ 1.S.226-232.
15. Varlamov O.O. Trudy NII radio. 2011. № 1. S. 34-44.
16. Podkosova Ja.G. Voprosy sovremennoj nauki i praktiki. Universitet im. V.I. Vernadskogo. 2011. № 2 (33). S. 104-111.
17. Podkosova Ja.G. Trudy Nauchno-issledovatel'skogo instituta radio. 2011. № 2. S. 5-16.
18. Materialy veb-sajta d.t.n. Varlamov O.O. http://www.ovar.narod.ru. 2012.
2. 19. Materialy veb-sajta kompanii MIVAR http://www.mivar.ru. 2012.
1. Varlamov O.O. Pattern Recognition and Information Processing (PRIP`2011): proceedings of the 11th International Conference (18 - 20 may, Minsk, Republic of Belarus). Minsk: BSUIR. 2011. P. 326-329.
2. Podkosova Y.G. Pattern Recognition and Information Processing (PRIP`2011): proceedings of the 11th International Conference (18 - 20 may, Minsk, Republic of Belarus). Minsk: BSUIR. 2011. P. 447-450.
3. Nosov A.V. Proceedings of the Congress on intelligent systems and information technologies "AIS-IT'10". Scientific publication in 4 volumes. Moscow: Physmathlit. 2010. Vol. 4. P. 118-119.
4. Pospelov D.A. Modelirovanie rassuzhdenij. Opyt analiza myslitel'nyh aktov. M.: Radio i svjaz'. 1989. 184 s.
5. Kuznecov O.P. Diskretnaja matematika dlja inzhenera. 6-e izd. SPb: Lan'. 2009. 400 s.
6. Kuk D. Komp'juternaja matematika. M.: Nauka. Fiz.-mat.lit. 1990. 384s.
7. Sajmon A.R. Strategicheskie tehnologii baz dannyh: menedzhment na 2000 god. M.: Finansy i statistika. 1999. 479 s.
8. Markov A.S. Bazy dannyh. Vvedenie v teoriju i metodologiju: Uchebnik. M.: Finansy i statistika. 2006. 512 s.
9. Kuznecov S.D. Bazy dannyh: jazyki i modeli. M.: OOO "Binom-Press". 2008. 720 s.
10. Kogalovskij M.R. Jenciklopedija tehnologij baz dannyh. M.: Finansy i statistika. 2005. 800 s.
11. Aho A.V. Struktury dannyh i algoritmy. M.: Izdatel'skij dom "Vil'jams". 2007. 400s.
12. Horoshevskij V.F. Iskusstvennyj intellekt i prinjatie reshenij. № 1. 2008. S. 80-97.
13. Ljuger Dzh.F. Iskusstvennyj intellekt: strategii i metody reshenija slozhnyh problem, 4-e izdanie. M.: Izdatel'skij dom "Vil'jams". 2005. 864 s.
3. Varlamov O.O. MIVAR: Transition from Productions to Bipartite Graphs MIVAR Nets and Practical Realization of Automated Constructor of Algorithms Handling More than Three Million Production Rules. http://arxiv.org/abs/1111.1321.

REZUME

O.O. Varlamov

MIVAR Technologies: Transition from Productions to Bipartite Graphs MIVAR Nets and Realization of Automated Constructor of Algorithms Handling More than Three Million Production Rules

The problem of elaboration of intellectual systems is up-to-date and important. Elaboration of expert systems of new generation would allow automation of solving difficult problems. The MIVAR (Multidimensional Informational Variable Adaptive Reality) approach has provided an opportunity to offer new models and methods for data mining and management. The first articles concerned some problems of graph theory and the elaboration of lineal matrix method of logic inference path finding on the adaptive net of rules [1-3]. Then, some work concerning the elaboration of MIVAR information space was done [4-5]. The most strictly formalized definition of the MIVARs can be found in papers [6-7]. Then the questions of the development [8-10] and the use of MIVARs for different simulators and instruction systems were discussed [11-22, 33]. MIVAR nets allow the creation of new "General Problem Solver", the prototype of which is an UDAV (Universal Designer Algorithms Varlamov) programme complex. MIVAR nets remove restrictions existed before and, in fact, create expert systems of new generation which are capable to process millions of rules

in acceptable time. MIVAR nets can be seen as a qualitative leap and transition to the new possibilities in information processing. The MIVAR approach unifies and develops achievements from different scientific domains such as databases, computational problems, logic processing, and includes two main technologies:

1) The MIVAR technology of information accumulation. It is the method of creation of global evolutional bases of data and rules (knowledge) with changeable structure based on the adaptive discrete MIVAR information space of unified representation of data and rules, which bases on three main definitions: “Thing, Property, Relation”.

2) The MIVAR technology of information processing. It is the method of creation of the system of logic inference or “automatic construction of algorithms from modules, services and procedures” based on the active MIVAR net of rules with lineal computational complexity.

The theoretical transition from the graphs of production systems to the bipartite graphs of the MIVAR nets is shown. Examples of the implementation of the MIVAR nets in the formalisms of matrixes and graphs are given. The linear computational complexity of algorithms for automated building of objects and rules of the MIVAR nets is theoretically proved. Different services, modules and computational procedures can be used as the MIVAR rules. The MIVAR method of data processing is fully described. It is proved that the bipartite graphs of the MIVAR nets can be used in the expert systems instead of productions and traditional graphs. On the basis of the MIVAR nets the UDAV software complex is developed, handling more than **1.17 million objects and more than 3.5 million rules** on ordinary computers. The results of experiments that confirm a linear computational complexity of the MIVAR method of information processing are given. UDAV can be used for the search of logic inference. MIVARs allow creating a “General Problem Solver”, the prototype of which is UDAV. Active logically thinking systems basing on the MIVARs can be created, which will be the prototype of Artificial Intelligence. In the perspective, a global multi-subject active expert system “The MIVAR Active Encyclopedia” will be created based on the MIVAR nets. The MIVAR nets eliminate previous limitations. MIVAR technologies of the data storage and processing expand computers technical possibilities and allow creating new intellectual systems. The MIVAR nets are the qualitative leap to the new possibilities of data processing.

Статья поступила в редакцию 02.07.2012.